

Consulta de un tabla completa

Elige los materiales

Algebra relacional.
materiales

SQL

select * from materiales

				Clave	Descripción	Costo
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar
<input type="checkbox"/>		Editar		Copiar		Borrar

Selección

Muestra la fila de la tabla de materiales donde la clave sea igual a 1000

Algebra relacional.

SL{clave=1000}(materiales)

SQL

select * from materiales

where clave=1000

				Clave	Descripción	Costo
<input type="checkbox"/>		Editar		Copiar		Borrar
				1000	Varilla 3/16	100

Proyección

Muestra las columnas clave, rfc y fecha de la tabla entregan
Algebra relacional.

$PR\{clave, rfc, fecha\}$ (entregan)

SQL

`select clave, rfc, fecha from entregan`

				clave	rfc	fecha
<input type="checkbox"/>		Editar		Copiar		Borrar
				1000	AAAA800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1000	AAAA800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1010	BBBB800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1010	BBBB800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1020	CCCC800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1020	CCCC800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1030	DDDD800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1030	DDDD800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1040	EEEE800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1040	EEEE800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1050	FFFF800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1050	FFFF800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1060	GGGG800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1060	GGGG800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1070	HHHH800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1070	HHHH800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1080	AAAA800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1080	AAAA800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1090	BBBB800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1090	BBBB800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1100	CCCC800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1100	CCCC800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1110	DDDD800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1110	DDDD800101	0000-00-00
<input type="checkbox"/>		Editar		Copiar		Borrar
				1120	EEEE800101	0000-00-00

Reunión Natural

Algebra relacional.

entregan JN materiales

SQL

```
select * from materiales,entregan
where materiales.clave = entregan.clave
Materiales entregados
```

Clave	Descripción	Costo	Clave	RFC	Numero	Fecha	Cantidad
1000	Vanilla 3/16	100	1000	AAAA800101	5000	0000-00-00	165
1000	Vanilla 3/16	100	1000	AAAA800101	5019	0000-00-00	254
1010	Vanilla 4/32	115	1010	BBBB800101	5001	0000-00-00	528
1010	Vanilla 4/32	115	1010	BBBB800101	5018	0000-00-00	523
1020	Vanilla 3/17	130	1020	CCCC800101	5002	0000-00-00	582
1020	Vanilla 3/17	130	1020	CCCC800101	5017	0000-00-00	8
1030	Vanilla 4/33	145	1030	DDDD800101	5003	0000-00-00	202
1030	Vanilla 4/33	145	1030	DDDD800101	5016	0000-00-00	295
1040	Vanilla 3/18	160	1040	EEEE800101	5004	0000-00-00	263
1040	Vanilla 3/18	160	1040	EEEE800101	5015	0000-00-00	540
1050	Vanilla 4/34	175	1050	FFFF800101	5005	0000-00-00	503
1050	Vanilla 4/34	175	1050	FFFF800101	5014	0000-00-00	623
1060	Vanilla 3/19	190	1060	GGGG800101	5006	0000-00-00	324
1060	Vanilla 3/19	190	1060	GGGG800101	5013	0000-00-00	692
1070	Vanilla 4/35	205	1070	HHHH800101	5007	0000-00-00	2
1070	Vanilla 4/35	205	1070	HHHH800101	5012	0000-00-00	503
1080	Ladrillos rojos	50	1080	AAAA800101	5008	0000-00-00	86
1080	Ladrillos rojos	50	1080	AAAA800101	5011	0000-00-00	699
1090	Ladrillos grises	35	1090	BBBB800101	5009	0000-00-00	73
1090	Ladrillos grises	35	1090	BBBB800101	5010	0000-00-00	421
1100	Block	30	1100	CCCC800101	5009	0000-00-00	466
1100	Block	30	1100	CCCC800101	5010	0000-00-00	699
1110	Megablock	40	1110	DDDD800101	5008	0000-00-00	337
1110	Megablock	40	1110	DDDD800101	5011	0000-00-00	368
1120	Sillar rosa	100	1120	EEEE800101	5007	0000-00-00	692

Si algún material no se ha entregado, ¿Aparecería en el resultado de esta consulta?

No, porque al igualar clave de materiales con clave de entrega se mostrarán únicamente los materiales que se hayan entregado, ya que si no se ha entregado no estaría en la tabla de entregados

Reunión con criterio específico

Selecciona los proyectos a los que se les entregaron

Algebra relacional.

entregan JN{entregan.numero <= proyectos.numero} proyectos

SQL

```
select * from entregan,proyectos
where entregan.numero <= proyectos.numero
```

Clave	RFC	Numero	Fecha	Cantidad	Numero	Denominacion Social
1000	AAAA800101	5000	0000-00-00	165	5000	Vamos Mexico
1000	AAAA800101	5000	0000-00-00	165	5001	Aztec?n
1000	AAAA800101	5000	0000-00-00	165	5002	5002
1000	AAAA800101	5000	0000-00-00	165	5003	Mexico sin ti no estamos completos
1000	AAAA800101	5000	0000-00-00	165	5004	Educando en Coahuila
1000	AAAA800101	5000	0000-00-00	165	5005	Infonavit Durango
1000	AAAA800101	5000	0000-00-00	165	5006	Reconstrucci?n del templo de Guadalup
1000	AAAA800101	5000	0000-00-00	165	5007	Construcci?n de plaza Magnolias
1000	AAAA800101	5000	0000-00-00	165	5008	Televisa en acci?n
1000	AAAA800101	5000	0000-00-00	165	5009	5009
1000	AAAA800101	5000	0000-00-00	165	5010	Construcci?n de Hospital Infantil
1000	AAAA800101	5000	0000-00-00	165	5011	Remodelaci?n de aulas del IPP
1000	AAAA800101	5000	0000-00-00	165	5012	Restauraci?n de instalaciones del CEA
1000	AAAA800101	5000	0000-00-00	165	5013	Reparaci?n de la plaza Sonora
1000	AAAA800101	5000	0000-00-00	165	5014	Remodelaci?n de Soriana
1000	AAAA800101	5000	0000-00-00	165	5015	CIT Yucatan
1000	AAAA800101	5000	0000-00-00	165	5016	Ampliaci?n de la carretera a la huast
1000	AAAA800101	5000	0000-00-00	165	5017	Reparaci?n de la carretera del sol
1000	AAAA800101	5000	0000-00-00	165	5018	Tu cambio por la educacion
1000	AAAA800101	5000	0000-00-00	165	5019	Queretaro limpio
1000	AAAA800101	5019	0000-00-00	254	5019	Queretaro limpio
1010	BBBB800101	5001	0000-00-00	528	5001	Aztec?n
1010	BBBB800101	5001	0000-00-00	528	5002	5002
1010	BBBB800101	5001	0000-00-00	528	5003	Mexico sin ti no estamos completos
1010	BBBB800101	5001	0000-00-00	528	5004	Educando en Coahuila
1010	BBBB800101	5001	0000-00-00	528	5005	Infonavit Durango
1010	BBBB800101	5001	0000-00-00	528	5006	Reconstrucci?n del templo de Guadalup
1010	BBBB800101	5001	0000-00-00	528	5007	Construcci?n de plaza Magnolias

Unión (se ilustra junto con selección)

Algebra relacional.

SL{clave=1450}(entregan) UN SL{clave=1300}(entregan)

SQL

(select * from entregan where clave=1450)

union

(select * from entregan where clave=1300)

←T→		Clave	RFC	Numero	Fecha	Cantidad
<input type="checkbox"/>	Editar	Copiar	Borrar	1300	GGGG800101	5005 0000-00-00 521
<input type="checkbox"/>	Editar	Copiar	Borrar	1300	GGGG800101	5010 0000-00-00 119

¿Cuál sería una consulta que obtuviera el mismo resultado sin usar el operador Unión?
Compruébalo.

```
SELECT *  
FROM entregan  
WHERE clave = 1450 OR clave = 1300;
```

Intersección (se ilustra junto con selección y proyección)

Algebra relacional.

$PR\{clave\}(SL\{numero=5001\}(entregan)) \cap PR\{clave\}(SL\{numero=5018\}(entregan))$

SQL

Nota: Debido a que en SQL server no tiene definida alguna palabra reservada que nos permita hacer esto de una manera entendible, veremos esta sección en el siguiente laboratorio con el uso de Subconsultas. Un ejemplo de un DBMS que si tiene la implementación de una palabra reservada para esta función es Oracle, en él si se podría generar la consulta con una sintaxis como la siguiente:



clave
1010

```
(select clave from entregan where numero=5001)  
intersect  
(select clave from entregan where numero=5018)
```

Diferencia (se ilustra con selección)

Algebra relacional.

$entregan - SL\{clave=1000\}(entregan)$

SQL

```
(select * from entregan)  
minus  
(select * from entregan where clave=1000)
```

Clave	RFC	Numero	Fecha	Cantidad
1010	BBBB800101	5001	0000-00-00	528
1010	BBBB800101	5018	0000-00-00	523
1020	CCCC800101	5002	0000-00-00	582
1020	CCCC800101	5017	0000-00-00	8
1030	DDDD800101	5003	0000-00-00	202
1030	DDDD800101	5016	0000-00-00	295
1040	EEEE800101	5004	0000-00-00	263
1040	EEEE800101	5015	0000-00-00	540
1050	FFFF800101	5005	0000-00-00	503
1050	FFFF800101	5014	0000-00-00	623
1060	GGGG800101	5006	0000-00-00	324
1060	GGGG800101	5013	0000-00-00	692
1070	HHHH800101	5007	0000-00-00	2
1070	HHHH800101	5012	0000-00-00	503
1080	AAAA800101	5008	0000-00-00	86
1080	AAAA800101	5011	0000-00-00	699
1090	BBBB800101	5009	0000-00-00	73
1090	BBBB800101	5010	0000-00-00	421
1100	CCCC800101	5009	0000-00-00	466
1100	CCCC800101	5010	0000-00-00	699
1110	DDDD800101	5008	0000-00-00	337
1110	DDDD800101	5011	0000-00-00	368
1120	EEEE800101	5007	0000-00-00	692
1120	EEEE800101	5012	0000-00-00	215
1130	FFFF800101	5006	0000-00-00	562

Nuevamente, "minus" es una palabra reservada que no está definida en SQL Server, define una consulta que regrese el mismo resultado.

```
(select * from entregan)
except
(select * from entregan where clave=1000)
```

en PHP my Admin

Producto cartesiano

Algebra relacional.
entregan X materiales

SQL

```
select * from entregan,materiales
```

Clave	RFC	Numero	Fecha	Cantidad	Clave	Descripción	Costo
1000	AAAA800101	5000	0000-00-00	165	1000	Varilla 3/16	100
1000	AAAA800101	5000	0000-00-00	165	1010	Varilla 4/32	115
1000	AAAA800101	5000	0000-00-00	165	1020	Varilla 3/17	130
1000	AAAA800101	5000	0000-00-00	165	1030	Varilla 4/33	145
1000	AAAA800101	5000	0000-00-00	165	1040	Varilla 3/18	160
1000	AAAA800101	5000	0000-00-00	165	1050	Varilla 4/34	175
1000	AAAA800101	5000	0000-00-00	165	1060	Varilla 3/19	190
1000	AAAA800101	5000	0000-00-00	165	1070	Varilla 4/35	205
1000	AAAA800101	5000	0000-00-00	165	1080	Ladrillos rojos	50
1000	AAAA800101	5000	0000-00-00	165	1090	Ladrillos grises	35
1000	AAAA800101	5000	0000-00-00	165	1100	Block	30
1000	AAAA800101	5000	0000-00-00	165	1110	Megablock	40
1000	AAAA800101	5000	0000-00-00	165	1120	Sillar rosa	100
1000	AAAA800101	5000	0000-00-00	165	1130	Sillar gris	110
1000	AAAA800101	5000	0000-00-00	165	1140	Cantera blanca	200
1000	AAAA800101	5000	0000-00-00	165	1150	Cantera gris	1210
1000	AAAA800101	5000	0000-00-00	165	1160	Cantera rosa	1420
1000	AAAA800101	5000	0000-00-00	165	1170	Cantera amarilla	230
1000	AAAA800101	5000	0000-00-00	165	1180	Recubrimiento P1001	200
1000	AAAA800101	5000	0000-00-00	165	1190	Recubrimiento P1010	220
1000	AAAA800101	5000	0000-00-00	165	1200	Recubrimiento P1019	240
1000	AAAA800101	5000	0000-00-00	165	1210	Recubrimiento P1028	250
1000	AAAA800101	5000	0000-00-00	165	1220	Recubrimiento P1037	280
1000	AAAA800101	5000	0000-00-00	165	1230	Cemento	300
1000	AAAA800101	5000	0000-00-00	165	1240	Arena	200

¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales?

Es igual a la multiplicación del número de tuplas de entrega (25) por el número de tuplas de materiales (25) lo que nos da un total de 625 tuplas

Construcción de consultas a partir de una especificación

Plantea ahora una consulta para obtener las descripciones de los materiales entregados en el año 2000.

Recuerda que la fecha puede indicarse como '01-JAN-2000' o '01/01/00'.

```
SELECT materiales.Descripción  
FROM materiales, entregan  
WHERE materiales.Clave = entregan.Clave  
and YEAR(entregan.Fecha) = 2000 or YEAR(entregan.Fecha) = 00;
```

Importante: Recuerda que cuando vayas a trabajar con fechas, antes de que realices tus consultas debes ejecutar la instrucción "set dateformat dmy". Basta con que la ejecutes una sola vez para que el manejador sepa que vas a trabajar con ese formato de fechas.

¿Por qué aparecen varias veces algunas descripciones de material?

Porque un material puede ser entregado muchas veces en un mismo año

Uso del calificador distinct

En el resultado anterior, observamos que una misma descripción de material aparece varias veces.

Agrega la palabra distinct inmediatamente después de la palabra select a la consulta que planteaste antes.

¿Qué resultado obtienes en esta ocasión?

Obtienes la descripción de los materiales unicos que fueron entregados en el año 2000

```
SELECT DISTINCT materiales.Descripción  
FROM materiales, entregan  
WHERE materiales.Clave = entregan.Clave  
AND (YEAR(entregan.Fecha) = 2000 OR YEAR(entregan.Fecha) = 2000);
```

Ordenamientos.

Si al final de una sentencia select se agrega la cláusula

order by campo [desc] [,campo [desc] ...]

donde las partes encerradas entre corchetes son opcionales (los corchetes no forman parte de la sintaxis), los puntos suspensivos indican que pueden incluirse varios campos y la palabra desc se refiere a descendente. Esta cláusula permite presentar los resultados en un orden específico.

Obtén los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas, ordenadas por número de proyecto, presentando las fechas de la más reciente a la más antigua.

```
SELECT proyectos.Numero as numeroProyecto,  
proyectos.DenominacionSocial as denominacionProyecto, entregan.Fecha  
as fechaDeEntrega, entregan.Cantidad as cantidadEntregada  
FROM proyectos, entregan  
WHERE proyectos.Numero = entregan.Numero  
ORDER BY proyectos.Numero ASC, entregan.Fecha DESC;
```

Uso de expresiones.

En álgebra relacional los argumentos de una proyección deben ser columnas. Sin embargo en una sentencia SELECT es posible incluir expresiones aritméticas o funciones que usen como argumentos de las columnas de las tablas involucradas o bien constantes. Los operadores son:

- + Suma
- Resta
- * Producto
- / División

Las columnas con expresiones pueden renombrarse escribiendo después de la expresión un alias que puede ser un nombre arbitrario; si el alias contiene caracteres que no sean números o letras (espacios, puntos etc.) debe encerrarse entre comillas dobles (" nuevo nombre"). Para SQL Server también pueden utilizarse comillas simples.

Operadores de cadena

El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado.

También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.

```
SELECT * FROM productos where Descripcion LIKE 'Si%'
```

¿Qué resultado obtienes?

Selecciona de la tabla productos todos los productos cuya descripción comience con SI

Explica que hace el símbolo '%'.
Es una especie de comodín el cual se utiliza cuando se quiere dar a entender que puede empezar o terminar de cierta manera

¿Qué sucede si la consulta fuera : LIKE 'Si' ?
Obtendrías los productos cuya descripción es de SI

¿Qué resultado obtienes?
La consulta estaría vacía

Explica a qué se debe este comportamiento.
A que la consulta buscara solamente los productos que su descripción sea exactamente SI

Otro operador de cadenas es el de concatenación, (+, +=) este operador concatena dos o más cadenas de caracteres.

Su sintaxis es : Expresión + Expresión.

Un ejemplo de su uso, puede ser: Un ejemplo de su uso, puede ser:

```
SELECT (Apellido + ', ' + Nombre) as Nombre FROM Personas;
```

```
DECLARE @foo varchar(40);  
DECLARE @bar varchar(40);  
SET @foo = '¿Que resultado';  
SET @bar = '¿¿¿???'  
SET @foo += ' obtienes?';  
PRINT @foo + @bar;
```

¿Qué resultado obtienes de ejecutar el siguiente código?
¿Que resultado obtienes? ¿¿¿???

¿Para qué sirve DECLARE?
Se utiliza para declarar una variable, su tipo y tamaño

¿Cuál es la función de @foo?
Es el nombre de la variable que le asignamos

¿Qué realiza el operador SET?
Asigna un valor a la variable previamente asignada

Sin embargo, tenemos otros operadores como [] , [^] y _.

[] - Busca coincidencia dentro de un intervalo o conjunto dado. Estos caracteres se pueden utilizar para buscar coincidencias de patrones como sucede con LIKE.

[^] - En contra parte, este operador coincide con cualquier caracter que no se encuentre dentro del intervalo o del conjunto especificado.

_ - El operador _ o guion bajo, se utiliza para coincidir con un caracter de una comparación de cadenas.

Ahora explica el comportamiento, función y resultado de cada una de las siguientes consultas:

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%';
```

Selecciona los RFC de los proveedores que realizaron entregas donde el RFC empieza con una letra entre A y D

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';
```

Selecciona los RFC de los proveedores que realizaron entregas donde el RFC no empieza con la letra A

```
SELECT Numero FROM Entregan WHERE Numero LIKE '____6';
```

Selecciona los RFC de los proveedores que realizaron entregas donde el RFC tenga 3 dígitos antes de un 6

Operadores compuestos.

Los operadores compuestos ejecutan una operación y establecen un valor.

+ = (Suma igual)

- = (Restar igual)

* = (Multiplicar igual)

/ = (Dividir igual)

% = (Módulo igual)

Operadores Lógicos.

Los operadores lógicos comprueban la verdad de una condición, al igual que los operadores de comparación, devuelven un tipo de dato booleano (True, false o unknown).

ALL Es un operador que compara un valor numérico con un conjunto de valores representados por un subquery. La condición es verdadera cuando todo el conjunto cumple la condición.

ANY o SOME Es un operador que compara un valor numérico con un conjunto de valores. La condición es verdadera cuando al menos un dato del conjunto cumple la condición.

La sintaxis para ambos es: valor_numerico {operador de comparación} subquery

BETWEEN Es un operador para especificar intervalos. Una aplicación muy común de dicho operador son intervalos de fechas.

```
SELECT Clave,RFC,Numero,Fecha,Cantidad  
FROM Entregan  
WHERE Numero Between 5000 and 5010;
```

¿Cómo filtrarías rangos de fechas?

```
SELECT Clave,RFC,Numero,Fecha,Cantidad  
FROM Entregan  
Where Fecha Between 1/01/2022 and 31/12/2022
```

EXISTS Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.
SELECT RFC,Cantidad, Fecha,Numero
FROM [Entregan]
WHERE [Numero] Between 5000 and 5010 AND
Exists (SELECT [RFC]
FROM [Proveedores]
WHERE RazonSocial LIKE 'La%' and [Entregan].[RFC] = [Proveedores].[RFC])

¿Qué hace la consulta?

Selecciona los datos de la tabla Entregan cuyo numero este entre 5000 y 5010 y que su razón social empiece con La

¿Qué función tiene el paréntesis () después de EXISTS?

Sirve para agrupar la subconsulta despues de exist

IN Especifica si un valor dado tiene coincidencias con algún valor de una subconsulta.

NOTA: Se utiliza dentro del WHERE pero debe contener un parametro. Ejemplo: Where proyecto.id IN Lista_de_Proyectos_Subquery

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador IN

```
SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Numero BETWEEN 2000 AND 5010
AND RFC IN (
    SELECT RFC
    FROM Proveedores
    WHERE RazonSocial LIKE 'La%'
)
```

NOT Simplemente niega la entrada de un valor booleano.

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador NOT IN Realiza un ejemplo donde apliques algún operador : ALL, SOME o ANY.

```
SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Numero BETWEEN 5000 AND 5010
AND RFC NOT IN (
    SELECT RFC
    FROM Proveedores
    WHERE RazonSocial LIKE 'La%'
);
```

El Operador TOP, es un operador que recorre la entrada, un query, y sólo devuelve el primer número o porcentaje específico de filas basado en un criterio de ordenación si es posible.

¿Qué hace la siguiente sentencia? Explica por qué.

```
SELECT TOP 2 * FROM Proyectos
```

Selecciona las primeras 2 líneas de la tabla proyectos porque Top 2 elige las primeras 2 líneas

¿Qué sucede con la siguiente consulta? Explica por qué.

Marcaría error porque Top espera un número específico y así no se llama a una variable en PHPMyAdmin o MySQL

```
SELECT TOP Numero FROM Proyectos
```

Modificando la estructura de un tabla existente.

Agrega a la tabla materiales la columna PorcentajeImpuesto con la instrucción:

```
ALTER TABLE materiales ADD PorcentajeImpuesto NUMERIC(6,2);
```

A fin de que los materiales tengan un impuesto, les asignaremos impuestos ficticios basados en sus claves con la instrucción:

```
UPDATE materiales SET PorcentajeImpuesto = 2*clave/1000;
```

esto es, a cada material se le asignará un impuesto igual al doble de su clave dividida entre diez.

Revisa la tabla de materiales para que compruebes lo que hicimos anteriormente.

¿Qué consulta usarías para obtener el importe de las entregas es decir, el total en dinero de lo entregado, basado en la cantidad de la entrega y el precio del material y el impuesto asignado?

```
SELECT e.Clave, e.RFC, e.Numero, e.Fecha, e.Cantidad,  
       m.Descripcion, m.Costo, m.PorcentajeImpuesto,  
       e.Cantidad * m.Costo * (1 + m.PorcentajeImpuesto / 100) AS ImporteTotal  
FROM Entregan e  
JOIN Materiales m ON e.Clave = m.Clave;
```

Creación de vistas

La sentencia:

```
Create view nombrevista (nombrecolumna1 , nombrecolumna2 ,..., nombrecolumna3 )  
as select...
```

Permite definir una vista. Una vista puede pensarse como una consulta etiquetada con un nombre, ya que en realidad al referirnos a una vista el DBMS realmente ejecuta la consulta asociada a ella, pero por la cerradura del álgebra relacional, una consulta puede ser vista como una nueva relación o tabla, por lo que es perfectamente válido emitir la sentencia:

```
select * from nombrevista
```

¡Como si nombrevista fuera una tabla!

Comprueba lo anterior, creando vistas para cinco de las consultas que planteaste anteriormente en la práctica . Posteriormente revisa cada vista creada para comprobar que devuelve el mismo resultado.

```
CREATE VIEW Descripciones_Entregas_2000 AS
SELECT m.Descripcion
FROM materiales
JOIN entregan ON materiales.Clave = entregan.Clave
WHERE YEAR(entregan.Fecha) = 2000 OR YEAR(entregan.Fecha) = 00;
```

```
CREATE VIEW Detalles_Entregas_Proyectos AS
SELECT proyectos.Numero AS NumeroProyecto, proyectos.Denominacion AS
DenominacionProyecto,
       entregan.Fecha AS FechaEntrega, entregan.Cantidad AS CantidadEntregada
FROM proyectos
JOIN entregan ON proyectos.Numero = entregan.Numero
ORDER BY proyectos.Numero ASC, entregan.Fecha DESC;
```

```
CREATE VIEW Detalles_Entregas_Proveedores_La AS
SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Numero BETWEEN 2000 AND 5010
AND RFC IN (
    SELECT RFC
    FROM Proveedores
    WHERE RazonSocial LIKE 'La%'
);
```

```
CREATE VIEW Detalles_Entregas_Sin_Proveedores_La AS
SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Numero BETWEEN 5000 AND 5010
AND RFC NOT IN (
    SELECT RFC
    FROM Proveedores
    WHERE RazonSocial LIKE 'La%'
);
```

```
CREATE VIEW Detalles_Importe_Entregas AS
```

```
SELECT e.Clave, e.RFC, e.Numero, e.Fecha, e.Cantidad,
       m.Descripcion, m.Costo, m.PorcentajeImpuesto,
       e.Cantidad * m.Costo * (1 + m.PorcentajeImpuesto / 100) AS ImporteTotal
FROM Entregan e
JOIN Materiales m ON e.Clave = m.Clave;
```

La parte (nombrecolumna1,nombrecolumna2,.de la sentencia create view puede ser omitida si no hay ambigüedad en los nombres de las columnas de la sentencia select asociada.

Importante: Las vistas no pueden incluir la cláusula order by.

A continuación se te dan muchos enunciados de los cuales deberás generar su correspondiente consulta.

En el reporte incluye la sentencia, una muestra de la salida (dos o tres renglones) y el número de renglones que SQL Server reporta al final de la consulta.

Los materiales (clave y descripción) entregados al proyecto "México sin ti no estamos completos".

```
SELECT m.Clave, m.Descripción
FROM Materiales m
JOIN Entregan e ON m.Clave = e.Clave
JOIN Proyectos p ON e.Numero = p.Numero
WHERE p.DenominacionSocial = 'Mexico sin ti no estamos completos';
```

Clave	Descripción
1030	Varilla 4/33
1230	Cemento
1430	Pintura B1022

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0005 segundos.)

Los materiales (clave y descripción) que han sido proporcionados por el proveedor "Acme tools".

```
SELECT m.Clave, m.Descripción FROM Materiales m
JOIN Entregan e ON m.Clave = e.Clave
JOIN Proveedores p ON e.RFC = p.RFC
WHERE p.RazonSocial = 'Acme tools';
```

Clave	Descripción
1020	Varilla 3/17
1020	Varilla 3/17
1100	Block
1100	Block
1180	Recubrimiento P1001
1180	Recubrimiento P1001
1260	Gravilla
1260	Gravilla
1340	Tuber?a 4.5
1340	Tuber?a 4.5
1420	Pintura C1012
1420	Pintura C1012

✓ Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0004 segundos.)

El RFC de los proveedores que durante el 2000 entregaron en promedio cuando menos 300 materiales.



```
SELECT e.RFC
FROM Entregan e
WHERE YEAR(e.Fecha) = 2000 or year(e.Fecha) = 00
GROUP BY e.RFC
HAVING AVG(e.Cantidad) >= 300;
```

←T→	RFC
<input type="checkbox"/> Editar Copiar Borrar	AAAA800101
<input type="checkbox"/> Editar Copiar Borrar	BBBB800101
<input type="checkbox"/> Editar Copiar Borrar	CCCC800101
<input type="checkbox"/> Editar Copiar Borrar	DDDD800101
<input type="checkbox"/> Editar Copiar Borrar	EEEE800101
<input type="checkbox"/> Editar Copiar Borrar	GGGG800101
<input type="checkbox"/> Editar Copiar Borrar	HHHH800101

✓ Mostrando filas 0 - 6 (total de 7, La consulta tardó 0,0004 segundos.)

El Total entregado por cada material en el año 2000.

```
SELECT e.Clave, SUM(e.Cantidad) AS TotalEntregado
FROM Entregan e
WHERE YEAR(e.Fecha) = 2000 or YEAR(e.Fecha) = 00
GROUP BY e.Clave;
```


<div><div><div></div><div></div><div></div></div></div>					Clave	TotalEntregado		
<input type="checkbox"/>		Editar		Copiar		Borrar	1000	419
<input type="checkbox"/>		Editar		Copiar		Borrar	1010	1051
<input type="checkbox"/>		Editar		Copiar		Borrar	1020	590
<input type="checkbox"/>		Editar		Copiar		Borrar	1030	497
<input type="checkbox"/>		Editar		Copiar		Borrar	1040	803

✓ Mostrando filas 0 - 24 (total de 44, La consulta tardó 0,0003 segundos.)

La Clave del material más vendido durante el 2001. (se recomienda usar una vista intermedia para su solución)

--crea la vista

```
CREATE VIEW MaterialMasVendido AS
SELECT Clave, SUM(Cantidad) AS TotalVendido
FROM Entregan
WHERE YEAR(Fecha) = 2001 or YEAR(Fecha) = 01
GROUP BY Clave;
```

-- hace la busqueda en la vista

```
SELECT Clave
FROM MaterialMasVendido
ORDER BY TotalVendido DESC
LIMIT 1;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0005 segundos.)

```
SELECT Clave FROM MaterialMasVendido ORDER BY TotalVendido DESC LIMIT 1;
```

Productos que contienen el patrón 'ub' en su nombre.

```
SELECT Descripción
FROM Materiales
WHERE Descripción LIKE '%ub%';
```

					Descripción
<input type="checkbox"/>		Editar			Recubrimiento P1001
<input type="checkbox"/>		Editar			Recubrimiento P1010
<input type="checkbox"/>		Editar			Recubrimiento P1019

✓ Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0003 segundos.)

Denominación y suma del total a pagar para todos los proyectos.

```
SELECT p.DenominacionSocial, SUM(e.Cantidad * m.Costo) AS TotalPagar
FROM Proyectos p
JOIN Entregan e ON p.Numero = e.Numero
JOIN Materiales m ON e.Clave = m.Clave
GROUP BY p.DenominacionSocial;
```

DenominacionSocial	TotalPagar
Ampliación de la carretera a la huast	565135
Aztecán	146595
5002	
CIT Yucatan	851700
Construcción de Hospital Infantil	144295

✓ Mostrando filas 0 - 17 (total de 18, La consulta tardó 0,0007 segundos.)

Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Solo usando vistas).

-- Crear vista de proveedores que suministran materiales al proyecto Televisa en acción

```
CREATE VIEW Proveedores_Televisa AS
SELECT DISTINCT p.RFC, p.RazonSocial
FROM Proveedores p
JOIN Entregan e ON p.RFC = e.RFC
JOIN Proyectos pr ON e.Numero = pr.Numero
WHERE pr.DenominacionSocial = 'Televisa en acción';
```

-- Crear vista final filtrando proveedores que no suministran al proyecto Educando en Coahuila

```
CREATE VIEW Proveedores_Televisa_No_Educando AS
SELECT pt.RFC, pt.RazonSocial
FROM Proveedores_Televisa pt
WHERE pt.RFC NOT IN (
    SELECT p.RFC
    FROM Proveedores p
    JOIN Entregan e ON p.RFC = e.RFC
    JOIN Proyectos pr ON e.Numero = pr.Numero
    WHERE pr.DenominacionSocial = 'Educando en Coahuila'
);
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0005 segundos.)

Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Sin usar vistas, utiliza not in, in o exists).

```
SELECT DISTINCT p.RFC, p.RazonSocial
FROM Proveedores p
JOIN Entregan e ON p.RFC = e.RFC
JOIN Proyectos pr ON e.Numero = pr.Numero
WHERE pr.DenominacionSocial = 'Televisa en acción'
AND p.RFC NOT IN (
    SELECT p2.RFC
    FROM Proveedores p2
    JOIN Entregan e2 ON p2.RFC = e2.RFC
    JOIN Proyectos pr2 ON e2.Numero = pr2.Numero
    WHERE pr2.DenominacionSocial = 'Educando en Coahuila'
);
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0014 segundos.)

Costo de los materiales y los Materiales que son entregados al proyecto Televisa en acción cuyos proveedores también suministran materiales al proyecto Educando en Coahuila.

```
SELECT m.Descripción, m.Costo, COUNT(e.Clave) AS CantidadEntregas,
SUM(m.Costo * e.Cantidad) AS TotalCosto FROM Materiales m JOIN
Entregan e ON m.Clave = e.Clave JOIN Proyectos pr ON e.Numero =
pr.Numero
JOIN Proveedores p ON e.RFC = p.RFC
WHERE pr.DenominacionSocial = 'Televisa en acción' AND p.RFC IN (
SELECT p2.RFC FROM Proveedores p2
JOIN Entregan e2 ON p2.RFC = e2.RFC JOIN Proyectos pr2 ON e2.Numero
= pr2.Numero WHERE pr2.DenominacionSocial = 'Educando en Coahuila' )
GROUP BY m.Descripción, m.Costo;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0027 segundos.)

Nombre del material, cantidad de veces entregados y total del costo de dichas entregas por material de todos los proyectos.

```
SELECT m.Descripción, COUNT(e.Clave) AS VecesEntregados, SUM(m.Costo
* e.Cantidad) AS TotalCosto
FROM Materiales m
JOIN Entregan e ON m.Clave = e.Clave
GROUP BY m.Descripción;
```

Descripción	VecesEntregados	TotalCosto
Arena	2	103600
Block	2	34950
Cantera amarilla	2	53590
Cantera blanca	2	160400
Cantera gris	2	1102310

✓ Mostrando filas 0 - 24 (total de 42, La consulta tardó 0,0007 segundos.)